AD-A247 888

March 10, 1992   Technical Report, Sep 16 - Dec 15, 1991
Quarterly

THE IMPACT OF ORGANIZATION STRUCTURE ON TEAM DECISION
MAKING UNDER STRESS AND UNCERTAINTY

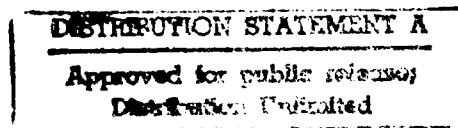Grant #: N00014-90-J-1680

R&T Project Number:
uri 5202-9002

Paul E. Lehner

Center of Excellence in Command, Control, Communications
 and Intelligence
George Mason University
Fairfax, Virginia 22030

GMU/C3I-115-IR

Office of Naval Research
Arlington, Virginia 22217-5000

unlimited

The objective of the research reported herein is to investigate coordination
in team decision making. Particular focus is placed on the identification
and characterization of variables that enhance coordination and enable teams
to maintain coordinated action under stressful conditions characteristic of
tactical environments.

DTIC
ELECTE
MAR 25 1992
S B D

Team Decision Making - Distributed Intelligence Systems -
Belief Networks - Petri Nets

32

unclassified   unclassified   unclassified   unclassified

CENTER OF EXCELLENCE IN
COMMAND, CONTROL, COMMUNICATIONS AND INTELLIGENCE

GEORGE MASON UNIVERSITY
Fairfax, Virginia   22030

QUARTERLY TECHNICAL REPORT

for the period

16 September 1991 - 15 December 1991

for

THE IMPACT OF ORGANIZATION STRUCTURE ON TEAM DECISION MAKING UNDER STRESS AND UNCERTAINTY

Grant Number N00014-90-J-1680
R&T Project Number uri 5202-9002

92-07619

1. PROGRAM OBJECTIVES

The objective of this research is to investigate coordination in hierarchical team decision making. Particular focus is placed on the identification and characterization of variables that enhance coordination and enable teams to maintain coordinated action under stressful conditions characteristic of tactical environments.

2. STATEMENT OF WORK

The research proposal identified three major tasks which define a sequence of three team decision making experiments. Each experiment involves the combined use of analytic models of the experimental setting and psychological models of human behavior to design the experiment and to predict performance.

Year 1 Experiment - The Year 1 experiment expands on the work of Jin (1990). The experiment will investigate the effects of time stress on team decision making performance. The experiment will be hosted on the testbed developed by Jin at MIT.

Year 2 Experiment - This experiment will focus on issues related to fixed versus variable structure organizations.

Year 3 Experiment - This experiment will extend the results of the previous experiments.

3. RESEARCH PLAN

The research plan describes our strategy for meeting the program objectives and fulfilling the research tasks. This research plan will evolve during the duration of this effort.

Our research plan has been organized into three highly related research areas:

(a)   Analytical models of C3I organizations that incorporate coordination variables,

(b)   Descriptive models of team decision making.

(c)   Prescriptive models of team decision procedures

The focus of the first area is the development of methodologies, models, theories and algorithms directed toward the derivation of superior tactical decision, coordination, and communication strategies of agents in organizational structures. Both fixed and variable organizational structures are considered. However, the focus is on modeling variable organizational structures and how those structures adapt under conditions of stress. The framework for this research is analytic.

The focus of the second area is the development of descriptive models of human decision making that are relevant to predicting team decision making performance under stress. For this work, it is assumed that the team members are well-trained. Consequently, the focus of the research is to identify conditions under which team performance degrades because one or more team members cannot effectively execute trained procedures properly.

The focus of the third area is to develop a prescriptive methdology for specifying team decision making procedures. This work will combine the normative and descriptive research in the first two areas to develop a methodology for deriving a set of robust team decision procedures. This

- 2 -

includes procedures for coordinating team decision making activities and adaptation of coordination procedures.

Each of the above areas is based on different scientific and engineering disciplines. It is our objective to merge these three research areas into a single theory of team design. We anticipate moving toward this objective through out the duration of this research program.

4. STATUS REPORT

In the context of the three tasks and research plan outlined above, a number of specific research problems have been formulated. These are being addressed by project faculty and by graduate assistants under the direction of project faculty. Each research problem is discussed below. Research problems which were completed during this period are described in some detail. Specifically, this progress report provides details on the general methodology for prescribing team ecision procedures (Research Task 4.3) and the results of the first experiment (Research Task +.4). A new task (Research Task 4.6) has been added.

4.1 CORDINATION IN DECISION MAKING ORGANIZATIONS

Background - The concept of an organization embodies two meanings. One is the physical entities and the interactions between them which form the organization. Another is the rules that govern the operation of the organization. We call all these physical entities and their interactions the system, and we characterize the operation of the system as coordination.

A key question in modeling and designing organizations is whether these two concepts can be decoupled. Mr. Zhuo Lu is investigating this problem under the supervision of Prof. Alexander H. Levis.

From the modeling point of view, if we can successfully decouple the organization model into two layers: the System Layer and the Coordination Layer, the modeling problem can be considered as two sub-problems: How to model the System Layer and how to model the Coordination Layer. The System Layer models all the system entities (including their interactions) with their built in functions; and the Coordination Layer models the rules of coordination for each system entity. Therefore, the modification of the Coordination Layer will not affect the System Layer. This may also provide a potential to formulate the organization by the mathematical description of System Layer and Coordination Layer.

From the designing point of view, the design problem may be divided into several sub problems:

- How to design the System Layer
- How to design the Coordination Layer given the System Layer
- How to modify the System Layer if the Coordination Layer is given

Monguillet and Levis (1988) initiated the investigation of variable structure Decision Making Organizations. Based on theory of Predicate Transition Nets (PTN) (Genrich, 1987), Monguillet extended the framework of System Effectiveness Analysis for comparing both variable and fixed organizations.

They also have refined the concept of variability. Three types of variability have been defined:

- Type 1-variable, if it adapts to the input it processes.
- Type 2-variable, if it adapts to environmental changes

- Type 3-variable, if it adapts to changes in the system's parameters

An appropriate mathematical framework is that of Colored Petri Nets (CPN) (Jensen, 1987), Demaël and Levis (1989) used it to investigate systems that adapt their structure of interactions to the input they process. Grevet and Levis (1988) have addressed the coordination problem in two classes of issues:

- The synchronization of the activity during the decision making processes.

- The consistency of the information processed by the different members of the organization.

In that work, a measure of coordination was introduced that depended on two other measures: synchronization and information consistency.

Progress to date:

During this period, the effort was focused on developing the detailed specification of the two-layered model of a component. The Coordination Layer depicts the rules which govern the operation of this component and contains the interactions with the Coordination Layer of other components. Therefore, supervisory coordination information can be exchanged between component models. Information received in the System Layer consists of the direct inputs to the system while information flows in the Coordination Layer are related to changes in the environment and system parameters.

Input information also contains coordination information. This part of the input information will flow to the Coordination Layer, as shown in Figure 4.1.1. The Coordination Layer will then generate controls to the system to govern the behavior of the component.
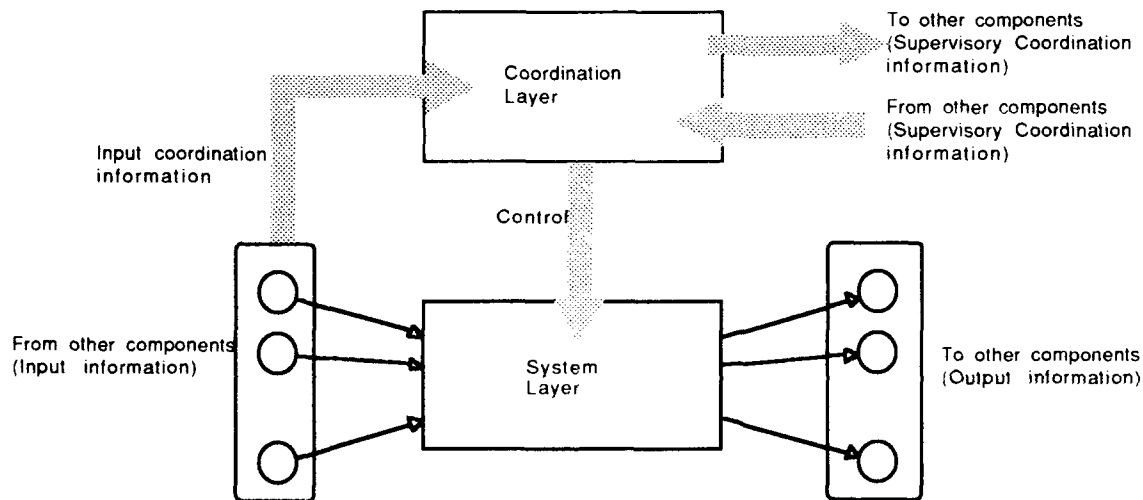

Figure 4.1.1   A component model

Figure 4.1.2 shows the System Layer of a component. The input places are P_1, P_2, ..., P_m. The transitions f_1, f_2, ..., f_n model the alternative functions the component can perform. The places O_1, O_2, ..., O_l are output places through which the component will send the output to other

components. The two places C_1 and C_2 control the behavior of the component, they act as the interface to the Coordination Layer.
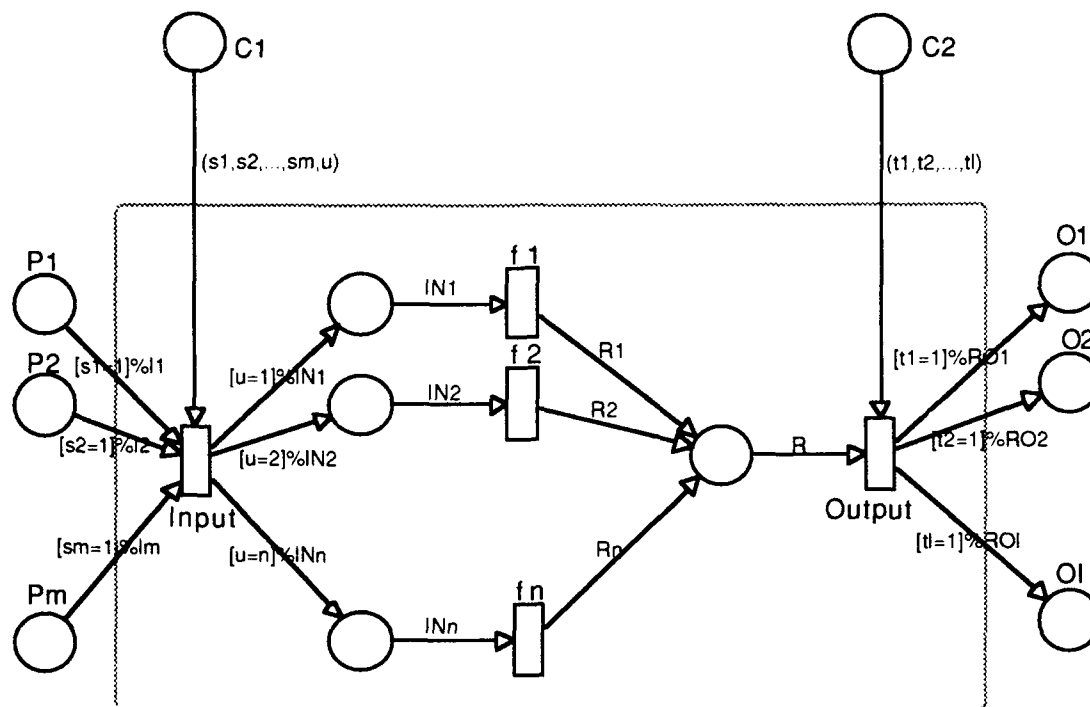

Figure 4.1.2   The System Layer of a component

The information in the input places of a component is a tuple I, which can be partitioned into two parts: $I = I_c + I_i$. I_c denotes the components of the tuple which relate to coordination, while I_i denotes the components of the tuple which do not relate to coordination. Only I_c will be sent to the Coordination Layer.

In Figure 4.1.3, the Coordination Layer is modeled using Colored Petri Nets. In the figure, CS1, CS2, ..., CSm are transitions with the Coordination Strategy embeded. The Coordination Layer can be separated into two layers:, the Supervisory Layer and the Executive Layer, based on the different roles they play in coordination.

The Executive Layer of a component consists of a set of Coordination Strategies CS1, CS2, ...,CSm. The information comes from the System Layer and the controls to the System Layer. The Supervisory Layer consists of the rules for selecting a Coordination Strategy to be executed, the controls to the Executive Layer and the exchanges of supervisory coordination information between different components.

This two-layered model of the component will be used as the building blocks for constructing two-layered models of organizations.

Documentation

Z. Lu and A. H. Levis, "A Colored Petri Net Model of Tactical Decision Making," *Proc. 1991 Symposium on C2 Research*, National Defense University, Ft. McNair, Washington, DC, June 1991.

Z. Lu and A. H. Levis, "A Colored Petri Net Model of Distributed Tactical Decision Making," to appear in *Proc. 1991 IEEE International Conference on Systems, Man, and Cybernetics*, October 1991.
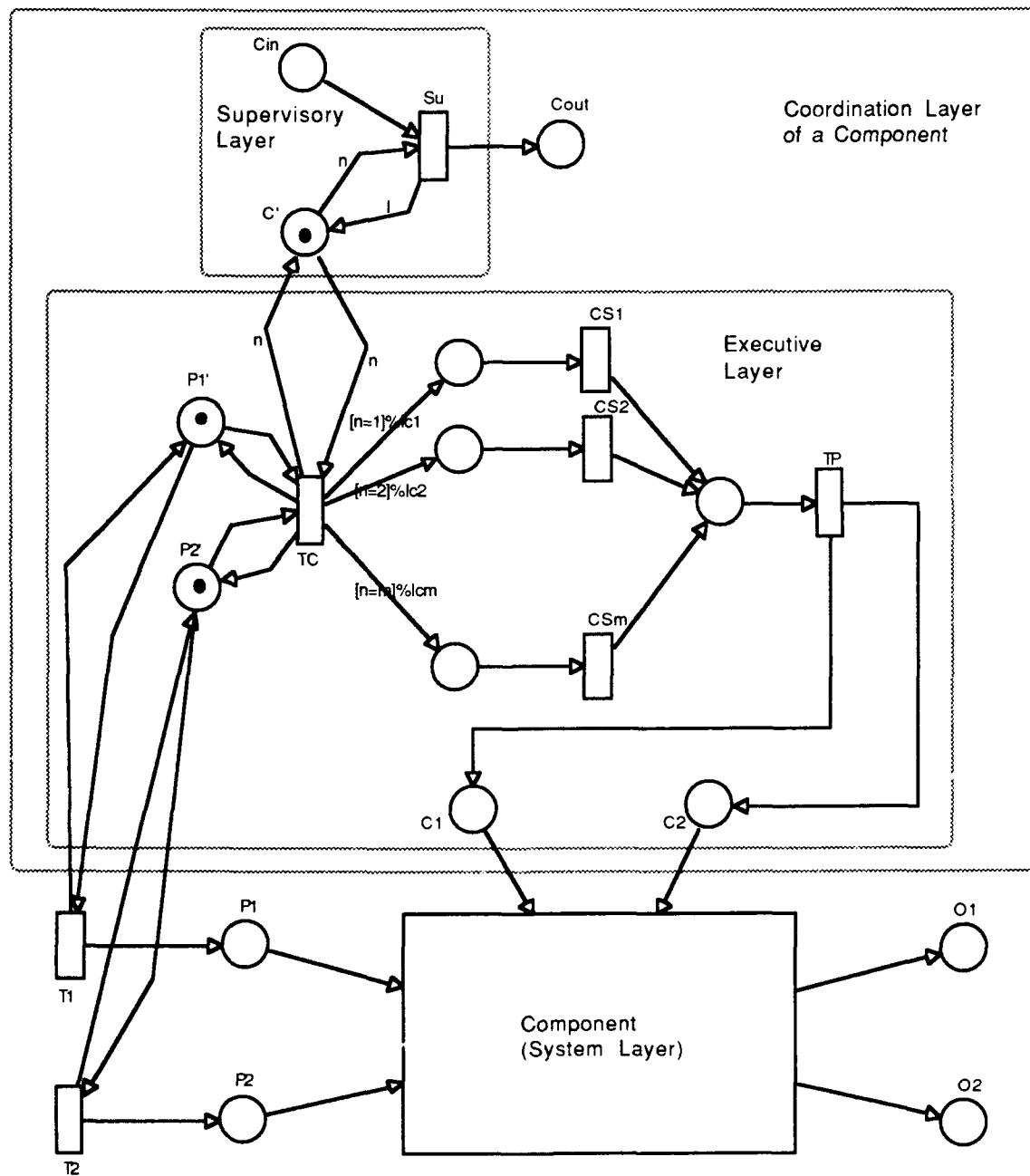

Figure 4.1.3   The Coordination Layer of a component

4.2 DESIGN OF MULTILEVEL HIERARCHICAL ORGANIZATIONS

Background - Both centralized and distributed organizations are characterized by the hierarchical structure. These organizational architectures are described by families of structures with each family concerned with the behavior of the organization as viewed from a different level of abstraction. Previous efforts under the Distributed Tactical Decision Making program resulted in a number of methodologies to design and generate *flat* architectures; the system is viewed only from a single level of detail (Remy and Levis, 1988). The basic decision making entity assumed throughout these methodologies was a human decision maker (DM). The current effort is directed towards a methodology to generate in some orderly manner, either by using the existing algorithms iteratively or by some new algorithm, the organizational structures for *multilevel* hierarchical organizations. This research task is necessary if realistic decision making organizations are to be modeled and analyzed. The research task is being carried out by Mr. Syed Abbas Zaidi under the supervision of Prof. Alexander H. Levis. This report first defines the problem and then describes the results obtained during this reporting period.

The following four issues must be addressed in order to implement such a methodology:

(a) The concept of multilevel hierarchical organizational structures needs to be formulated analytically.

(b) A mathematical framework that is appropriate for the formulation of the design problem should be identified.

(c) Sets of constraints have to be identified for different levels in the organization to reflect design requirements and to keep the problem of generating organizational structures computationaly feasible.

(d) A set of connectivity rules needs to be formulated in order to integrate organizational structures defined at different degrees of abstraction.

Multilevel Hierarchical Systems

The concept of a multilevel, hierarchical system is defined in Mesarovic et al. (1970). Some of the characteristics which every hierarchy has are: *vertical arrangement of subsystems which comprise the overall system, priority of action or right of intervention of the higher level subsystems, and dependence of the higher level subsystems upon actual performance of the lower level.*

Mesarovic et al. (1970) defined three types of hierarchical systems. The classification is based on three notions of levels:

* The level of description or abstraction, the *stratum*.
* The level of decision complexity, the *layer*.
* The organizational level, the *echelon*.

The term *level* is reserved as a generic term referring to any of these notions when there is no need to distinguish between them.

The concept of stratum is used for modeling organizational architectures when viewed from different levels of abstraction, while the concept of layer is introduced in reference to the vertical

decomposition of a decision problem into sub-problems. The concept of echelon refers to the mutual relationship between Decision Making Units (DMU) comprising a system.

It is necessary to make a clear distinction as to which notion of level one is using when describing a hierarchical system. The type of multilevel, hierarchical systems under investigation are *stratified systems*, where the system is described by a family of structures each concerned with the behavior of the system as viewed from a different level of abstraction, the *stratum*. A Stratified Decision Making Organization is defined formally as follows:

A *Stratified Decision Making Organization (SDMO)* is defined to be a Decision Making Organization (DMO) in which a system on a given stratum is a subsystem on the next higher stratum. In a SDMO, DMUs can be either Decision Making Sub-Organizations (DMSO) or human Decision Makers (DM) depending upon the level of abstraction used to represent the organizational structure of the DMO.

For illustrative purposes, a description of a general SDMO is presented in Figure 4.2.1.
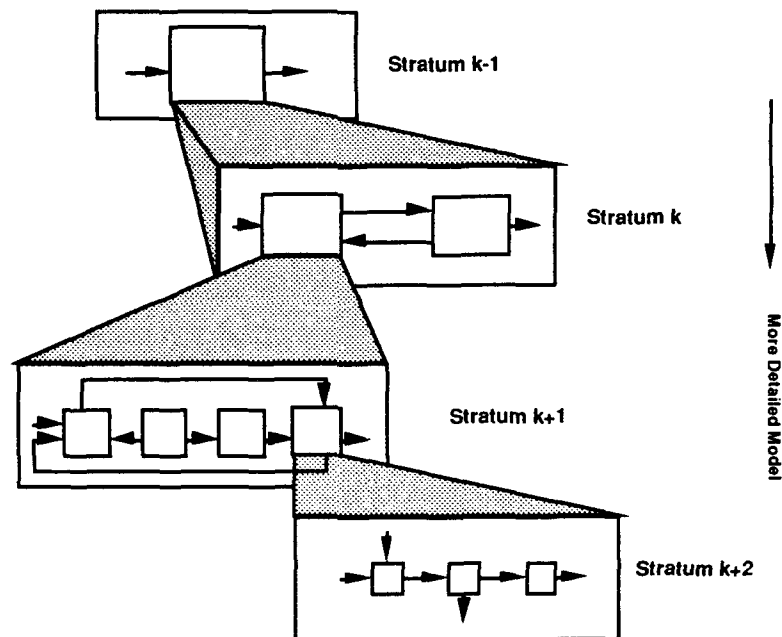

Figure 4.2.1   A Stratified Decision Making Organization (SDMO)

In a SDMO, the highest stratum, stratum '0', contains only one organizational structure, the *node*, which represents the entire organization (SDMO). The nodes at all other strata are referred to as Decisiob Making Units or DMUs. The node at stratum '0' shows the highest level of abstraction that can be used to describe an organizational structure. On the other hand, the nth stratum contains an elaborated and detailed description of the DMO at the lowest level of abstraction that is determined by the designer of the organization. The range of 'n' is defined as $1 \leq n \leq N$, where 'N' represents the lowest possible stratum at which the DMUs cannot be decomposed further. The determination of the value of 'N' is application dependent, i.e., it depends upon the kind of organization being modeled, and on the definition of strata used to describe the organization. For example, in human organizations, 'N' represents the stratum at which the DMUs are individual human decision makers (DMs)

A DMU at stratum 'k', where $1 \leq k < n$, is defined as a *compound node* to reflect the Petri Net formalism that is used to describe the organizational structure. All nodes are labeled by an alphanumeric code, $\mathbf{DMU_{ik}}$, where 'i' represents the node number at stratum 'k'. The set of all the nodes at stratum 'k' contains $|\mu_k|$ elements, i.e.,

$$\mu_k = \{1, 2, ..., |\mu_k|\} \text{ and } i \in \mu_k.$$

The following property holds for every stratified decision making organization (SDMO): The number of nodes at a stratum is larger than or equal to the number of nodes in the stratum immediately above it.

$$|\mu_n| \geq |\mu_{n-1}| \geq ... \geq |\mu_{k+1}| \geq |\mu_k| \geq |\mu_{k-1}| \geq ... \geq |\mu_0| = 1 \qquad\qquad 1 \leq n \leq N$$

This follows from the fact that a system on a stratum is comprised of a number of subsystems which are defined for the next lower stratum; the number of nodes at a given stratum is given by the sum of the subsystems of the individual nodes at the next higher stratum.

Progress to Date

The previous report outlined the findings of research done on the first three of the above mentioned issues. This report focuses on the resolution of the connectivity problem.

The problem of interpreting higher level interactions in terms of their lower level representation arises when an organizational structure is unfolded to its lower level description. The problem of connectivity is illustrated in Figure 1, where an organizational structure in stratum 'k' is unfolded to its stratum 'k+1' representation. A set of connectivity rule is needed to translate the interaction represented by P_{2ijk}, Figure 4.2.2, to its lower level representation. All the shaded arcs in Figure 1(b) show the candidate input/output connections that can represent the lower level connectivity of the interaction, P_{2ijk}.
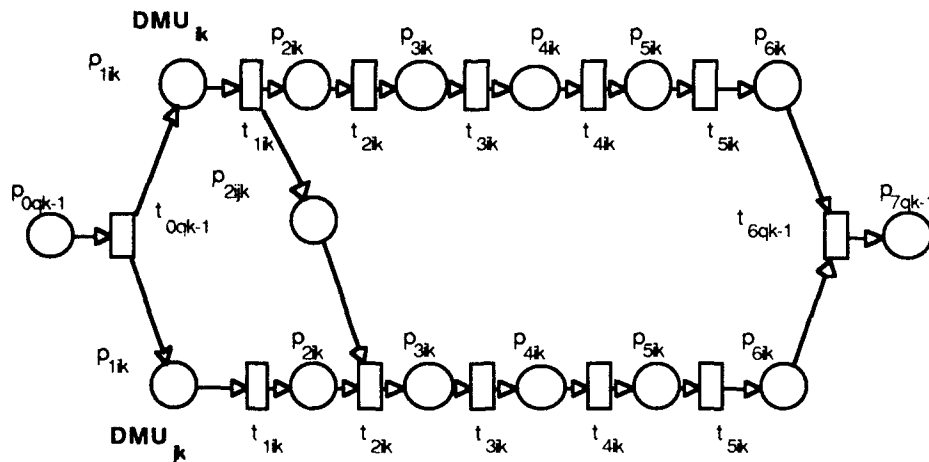
(a) Stratum 'k' Representation
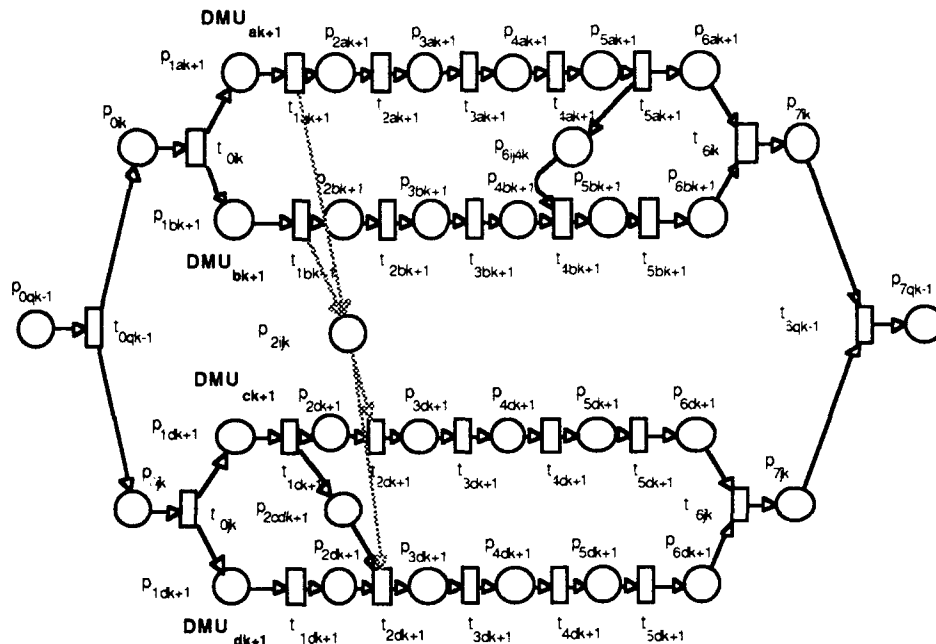
(b) Stratum 'k+1' Representation

Figure 4.2.2 Connectivity Problem

SOLUTION TO THE DESIGN PROBLEM

A set of connectivity rules was formulated to resolve the connectivity problem. The connectivity rules are based upon the multiechelon hierarchical relationship that may exist among the DMUs of an organizational structure. In order to define the multiechelon hierarchy among organizational members, the *messages* that flow in an organization are classified into different categories. These categories are shown in Figure 4.2.3. An ordering was then defined on input and output messages to characterize the echelon type relationship among different organizational members. Table 4.2.1 represents the ordering in terms of input and output messages. In defining the ordering, it is assumed that the DMU with one of the input (output) messages, listed in Table 4.2.1, is considered isolated from all other output(input) messages, it may or may not have.
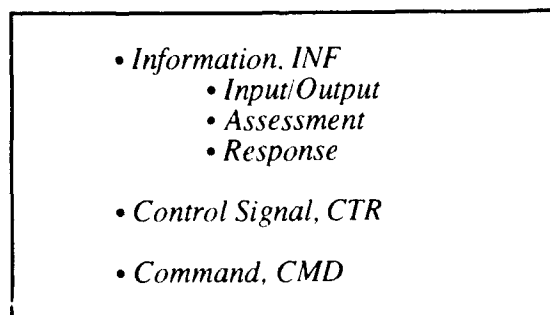

- *Information. INF*
  - *Input/Output*
  - *Assessment*
  - *Response*

- *Control Signal, CTR*

- *Command, CMD*

Figure 4.2.3 Classification of Messages

A DMU is characterized as a 2-tuple, (I, O), where 'I' corresponds to the order defined by the input interactions of the DMU, and 'O' represents the order defined by the output interactions of the DMU, Table 4.2.1. The set of all the elements of the matrix is represented by Π. It is defined to be the set of DMUs with all the possible interactional structures and their associated input and output ordering.An echelon index was then defined based on the orderings described for input and output messages. The lattice structure of Π and the echelon index are presented in Figure 4.2.4.

Table 4.2.1   Ordering in Terms of Input and Output messages

Interactions	Corresponding Order on Inputs	Corresponding Order on Outputs
INF	1	7
INF, CTR	2	6
CTR	3	5
INF, CMD	4	4
INF, CTR, CMD	5	3
CTR, CMD	6	2
CMD	7	1
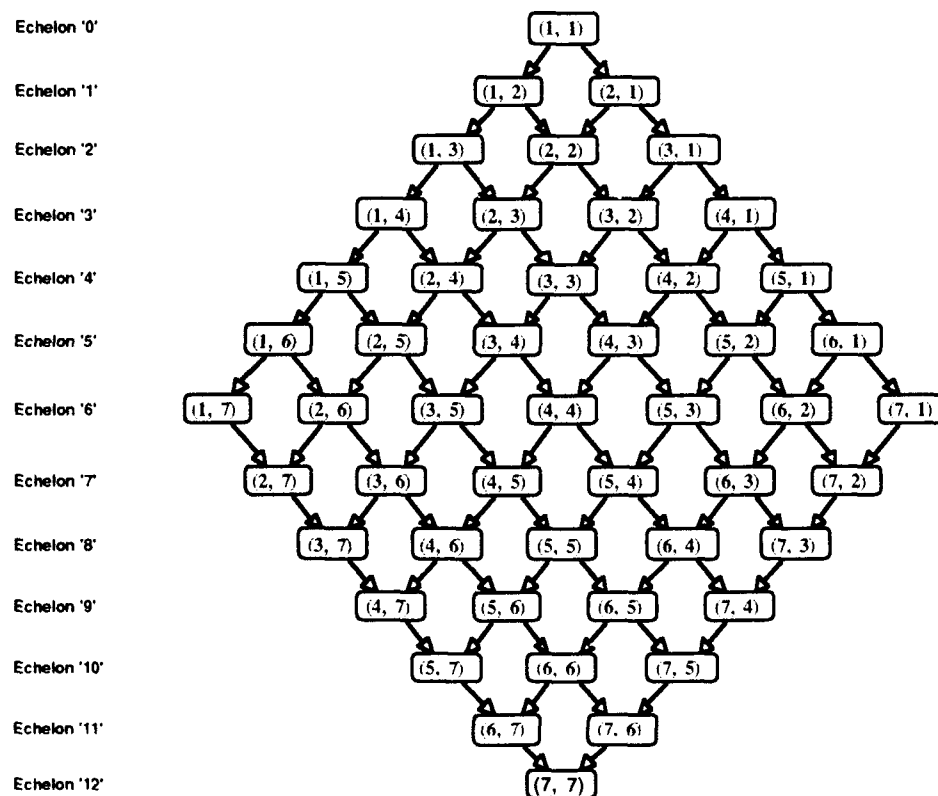
Echelon Index = I + O -2

Figure 4.2.4   Multiechelon Hierarchy

According to the methodology being proposed, whenever it is desired to translate a higher stratum interactional link to its lower stratum description, all the subsystems defined in the lower stratum are identified in terms of their echelon indices and then the connectivity rules are applied to the structure. The connectivity rules being formulated are illustrated as follows.

Let us consider two compound nodes 'i' and 'j' in stratum 'k'. The compound nodes 'i' and 'j' themselves are DMUs of an organizational structure 'q' defined at stratum 'k-1'. The subsystems (DMUs) of compound node 'i' are given as 'a' and 'b', while compound node 'j' is composed of DMUs 'c' and 'd', Figure 7.4 (reproduction of Figure 5.7). The rules of connectivity now can be formulates as follows:

• Rule 1

> An interactional link defined at stratum 'k' from a compound nodes 'i' to another compound node 'j' is translated into a single link at stratum 'k+1' from DMU 'a' or 'b' to DMU 'c' or 'd'.

• Rule 2

> The translated lower stratum interactional link between the subsystems of the compound nodes 'i' and 'j' will connect the highest echelon-DMUs of the two suborganizational structures. The highest echelons identified for the subsystems of 'i' and 'j' need not necessarily be the same.

• Rule 3

> If a compound node has two or more DMUs at the same highest echelon, the following rule applies:
>
> > • For an output interaction the DMU with higher 'O' index is selected
> > • For an input interaction the DMU with higher 'I' index is selected.
> > • For two or more DMUs with identical (I, O) indices, one of them is selected arbitrarily.

• Rule 4

> If in following Rules 1 to 3 constraint R2 is violated, then the next highest echelon-DMU will be selected to participate in the interaction. The identification of the next highest echelon-DMU follows the procedure presented in Rules (2) and (3).

THE DESIGN ALGORITHM

The algorithm for generating Stratified Decision Making Organizational (SDMO) structures. was developed by connecting the concepts and results presented in the previous and the present reports. Figure 4.2.5 presents the Bottom-Up approach of the design methodology. The Top-Down approach is equivalent to the Bottom-Up approach in the sense that the results produced are identical. In case the emphasis of the designer is on the subsystems of the entire organization then the Bottom-Up approach should be adopted in order to generate the organizational structure. On the other hand, if the emphasis is on the entire system instead of its subsystems and the manner in which it evolves, then the Top-Down approach is the one to be considered. In this approach, the design process starts with the entire system designed at the highest abstraction (highest stratum description) desired. then each subsystem is modeled in terms of its higher stratum description. Therefore. the design procedure produces information about the system at a relatively higher degree of abstraction as the process evolves. A detailed comparative study of the two approaches. however, will be presented in the technical report.

Step 0: *Initialization*

Specify total number of strata in the organization description,
$k = n$ (or possibly N). Specify DMUs in stratum 'k', μ_k, Go to Step 1.

Step 1:

Specify DMUs in stratum 'k-1', μ_{k-1}. If $k-1 = 0$ then $|\mu_{k-1}| = 1$. Map μ_k to μ_{k-1}; determine the elements of the set M:

$$\forall\ i \in \mu_{k-1} \quad \text{identify the set } M_{ik-1}$$

$M_{i\ k-1} = \{\ DMU_{jk} / j \in \mu_k,\ \exists\ a\ DMU_{k-1}\ i \in \mu_{k-1}\ ,\ \text{s.t. } DMU_{jk} \text{ is a subsystem of } DMU_{k-1}\ \}$

where

$$\sum_{i=1}^{|\mu_{k-1}|} |M_{ik-1}| = |\mu_k| \qquad k = 1, 2, ..., n$$

Go to Step 2.

Step 2: *Lattice Algorithm*

$\forall\ i \in \mu_{k-1}$, apply Lattice algorithm to generate Feasible Organization structures for DMU_{ik-1} with $|M_{ik-1}|$ decision making entities. Select one of the candidate structures, Σ_{ik}, and store its incidence matrix $\Delta_{i,\ k-1,\ k}$.
If value of the counter k is equal to 1, go to Step 3. Otherwise, decrement the counter by 1, $k = k-1$, and go to Step 1.

Step 3:

Specify stratum 'l' for which the system's description of a node, DMU_{js}, $s = 0, 1, ..., n$, is required. A general constraint on 'l' is given by:

$$s + 2 \leq l \leq n$$

For $l = s$, the system description is depicted by the compound node structure of DMU_{js}. STOP.
For $l = s + 1$, the required description is readily available after the application of the Lattice algorithm (Step 2) on DMU_{js}. Go to Step 6.
If l satisfies the constraint, set $u = s + 2$. Go to Step 4 with the description Σ_{js+1} and $\Delta_{j,\ s,\ s+1}$ of DMU_{js}.

Step 4: *Unfolding*

Unfold the present description of DMU_{js} to stratum 'u' according to the procedures outlined. Go to Step 5.

Figure 4.2.5   Design Algorithm

The proposed algorithm was applied to a number of design problems. The results of this effort show that the methodology and the resulting models provide a structured and modular way for solving the problem of designing large-scale distributed intelligence systems by breaking a computationaly large problem into smaller problems. The result also show a significant reduction in the computational effort required to generate the feasible solutions.

Documentation

Levis, A. H., "A Colored Petri Net Model of Intelligent Nodes" *Proc. 1991 IMACS Symposium on Modeling and Control of Technological Systems,* Lille, France, May 1991. The paper was also selected to appear in a book by Elsevier - North Holland with selected reprints from several IMACS symposia. Book to appear in 1992.

Levis, A. H., "A Colored Petri Net Model of Command and Control Nodes" to appear in *Command, Control, and Communications: Advanced Concepts and Paradigms,* Carl R. Jones, Ed., AIAA Press, Washington DC. Book to appear in late 1992.

Zaidi, S. A. K., "On the generation of Multilevel, Distributed Intelligence Systems using Petri Nets," M.S. Thesis, Report No.: GMU/C3I-113-TH, C3I Center, George Mason University, Fairfax, VA. November 1991.

4.3   GENERAL METHODOLOGY FOR PRESCRIBING TEAM DECISION PROCEDURES.

Background - A team is a well-trained group of decision makers with overlapping areas of expertise. Each team member has an area of responsibility, a set of decision functions for which that team member is responsible, and a protocol for communicating with other team members. Previous work in the mathematical modeling of teams has addressed the problem of specifying organizational structures, but there has been very little work addressing the problem of specifying the procedures embedded in each decision function.

In team decision making, a function corresponds to a set of decision procedures. For instance, a team member may be responsible for the function Interpert-Sensor-Readings, where it is the team member's responsibility to read a set of sensor displays (input) and to report values for

Probable-Current-Situation (output). Another team member may be responsible for the function Allocate-Air-Resources, where it is the team member's responsibility to use information about the Probable-Current-Situation to determine how to allocate air resources.

The performance of a team depends on the decision procedures each team member has been trained to execute and how effectively and reliably those procedures are executed. The objective of this research activity is to develop an approach to prescribing a set of decision procedures that (a) will lead to high performance, and (b) team members can reliably execute even under conditions of high stress.

Progress to date. A general approach for prescribing team decision procedures is described below In its current state of development, this approach emphasizes the use of probability models for modeling decision situations. This description is more detailed than provided in the previous progress report.

In its current state of development, the approach focuses on the use of probability models for modeling decision situations. Utility information is not fully considered. Further work will occur in this area.

Some of the specific steps within this methodlogy are being addressed as seperate research tasks. Specifically, research task 4.4 addresses the problem of determining the impact of cognitive biases, research task 4.5 addresses the problem of network compilation, and research task 6 addresses the related problem of incoporating decision aids into the team decision making process.

Overview of Approach - Our approach to prescribing team decision procedures is depicted in Figure 4.3.1. Each of these steps are described below.

Develop Domain Model - The initial step in this process is that of developing a domain model. Domain models are developed using a formalism called a *probability network*.[1] A probability network provides both a graphic and functional description of the probabilistic relationships between objects and events in a problem domain (Neopolitan, 1990).

To illustrate the use of probability networks, consider the situation depicted in Figure 4.3.2. It depicts an outer air battle scenario where the objective of a C^2 team is to identify the type and intent of each detected airborne track (the black arrows) before these tracks cross the outer circle. Each track may be composed of multple aircraft. The C^2 team must make this judgement based on data available from two passive sensor systems and a radar system. The passive sensors will report the apparent level of electronic emissions coming from each track (strong emmisions, weak emmisions, no emmisions). The radar system will report the apparent direction, speed and size (# of aircraft) of each track. All three sensors are subject to faults, and the sensor readings are probabilistically related to the true values. Futhermore, if a track is jamming, then the radar image around the area of that track will be snowy. Finally, for both kinds of sensors, the accuracy of the readings decrease with distance.[2]

---

[1] Probability networks are also referred to as belief networks, Bayesian belief networks and inference networks.
[2] This is not intended to be a "real world" example. There is, for instance, no mention of IFF signals and procedures.
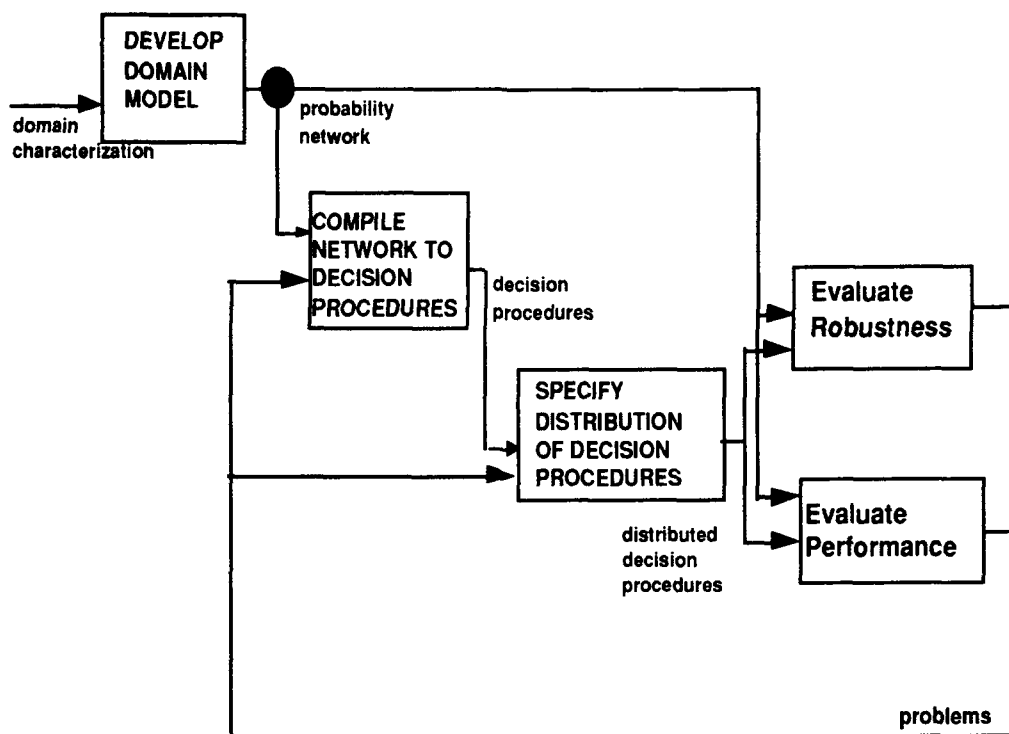
Figure 4.3.1 Overview of Prescriptive Theory

Figure 4.3.3 depicts a probability network that shows the relationships between the entities in this domain. The probabilistic distribution of a node conditioned on its parents is maintained a set of conditional probability statements attached to each node.

There are a variety of *Bayesian update algorithms* or *probability propagation algorithms* that can be used to process a probability network (Neopolitan, 1990). Any of these algorithms can be used to calculate the probability distribution of any node conditioned on specific values for any subset of the other nodes. For instance, conditioned on the following information

general hostilities	=	*open hostility*
passive sensor 1	=	*strong emisions*
passive sensor 2	=	*weak emisions*
radar image	=	*snowy*

probability propagation could be used to calculate the posterior probability that **intent** = *hostile*.[3,4]

---

[3]Given the existence of such algorithms, it is conceivable that the team decision process could be completely automated. We are assuming, however, that for whatever reason a policy decision was made to allocate the identification task to human decision makers.

[4]These probability statements represent statements about the expected relative frequency of events in an operational scenario. They do not necessarily represent what the situation is expected to be, but characterize instead the types of situations the team should be able to handle competently.
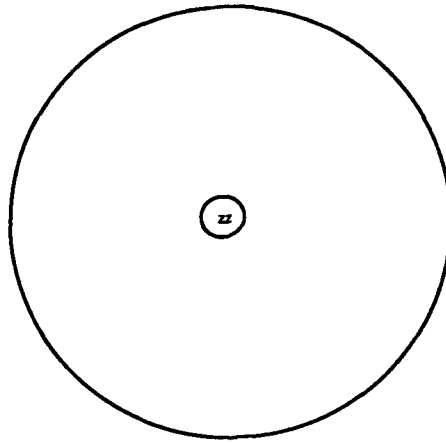
Figure 4.3.2 Depiction of Sample Decision Problem

Develop Decision Procedures - The second step in the process is to propose a set of decision procedures that can be used by each team member. There are two general considerations that are relevant to determing these decision procedures: performance and bounded rationality. If bounded

Figure 4.3.3 Example Domain Model using a Probability Network Representation

rationality constraints were not an issue. then one could simply require that the team use Bayesian updating to calculate posterior probabilities. This, however, is humanly impossib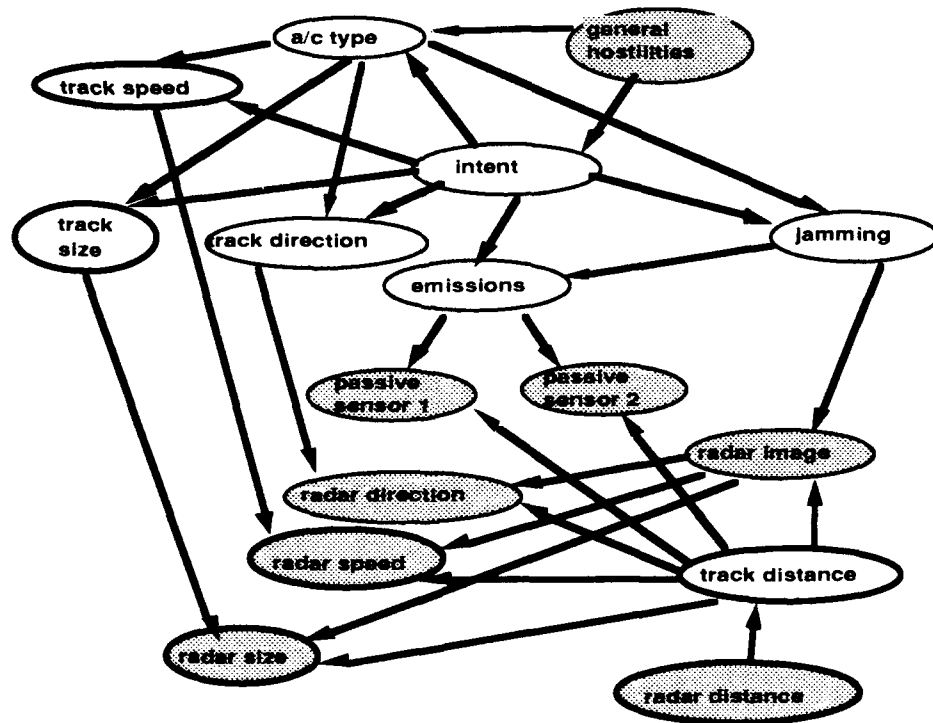le. --- Instead, one must find decision procedures that are easier to execute, but approximate the performance of Bayesian updating. In effect, it is necessary *compile the domain model* into a set of humanly executable decision procedures.

The example below shows how probability propagation algorithms can be used to support specification of a set of decision procedures.

Specify Context States - In real world problems, team decision making procedures are often conditioned to fit a background context. For instance, decision procedures for handling an unidentified track may differ greatly depending on whether or not the background situation is hostile. From the perspective of probability propagation there is no difference between context information (e.g., current state of hostilities) and problem-specific evidence (e.g., sensor readings from a specific track). However the distinction is useful for deciding how to compile the network. In the case of Figure 4.3.3, we will use the current status of general hostilities as background context.

Propose and Test Individual Rules - The probability network provides the team designer with an ability to test the behavior of individual rules. A team designer might know, for instance, that one common attack scenario is for a track to come in fast and jamming. Consequently, for a context where **general hostilities** = *hostile* the team designer might propose the following rule:

IF OBSERVE   R1
  passive sensor 1   = *strong emissions*   **AND**
  passive sensor 2   = *strong emissions*   **AND**
  radar image   = *snowy*   **AND**
  radar speed   = $\geq 650nm$
THEN DEDUCE
  intent   = *hostile.*

To test this rule, one could invoke a probabilistic processing algorithm to asses that

Prob(intent = *hostile.*| P1
  passive sensor 1   = *strong emissions*   **&**
  passive sensor 2   = *strong emissions*   **&**
  radar image   = *snowy*   **&**
  radar speed   = $\geq 650nm$
  general hostilities = *hostile*   )   = .94.

In addition, one could use probability propagation to explore simplifications of this rule. For example, one can perform value of information (VOI) calculations to show the change in posterior probability due to individual preconditons in a rule. In the above rule, for instance, one might wonder if **radar speed** is a useful precondition since a snowy radar image would imply that the reading of speed is unreliable. This can be determined by checking the probabilities

Prob(intent = *hostile.*| P2
  passive sensor 1   = *strong emissions*   **&**
  passive sensor 2   = *strong emissions*   **&**
  radar image   = *snowy*
  general hostilities = *hostile*   )   = .92.

Prob(intent = *hostile.*| P3

        passive sensor 1   = *strong emissions*   **&**
        passive sensor 2   = *strong emissions*   **&**
        radar image   = *snowy*   **&**
        radar speed   = *≤350nm*
        general hostilities = *hostile*   )   = .88.

P2 indicates that **radar speed** was not a particularly informative, since dropping this precondition did not substantially change the conditional probability. Futhermore, P3 tells us that even if the reading for **radar speed** is exactly opposite of what would be expected in a "fast and jamming" scenario, the posterior probability of **intent** = *hostile*.remains nearly the same. Consequently, **radar speed** can be safely dropped as a precondition.

To carry this example further, the team designer might wonder whether it is necessary to have both passive sensors report *strong emissions*. He or she might note that

Prob(intent = *hostile.*| P4

        [passive sensor 1   = *strong emissions* ∨
         passive sensor 2   = *strong emissions*]
        radar image   = *snowy*
        general hostilities = *hostile*   )   = .89.

and

Prob(intent = *hostile.*| P5

        passive sensor 1   = *strong emissions*   **&**
        passive sensor 2   = *no emissions*   **&**
        radar image   = *snowy*
        general hostilities = *hostile*   )   = .23.

P4 statement suggests that on an average it is only necessary to have one passive sensor report *strong emissions* . Consequently the following rule seems valid

IF OBSERVE   R2

        [passive sensor 1   = *strong emissions*   **OR**
              passive sensor 2   = *strong emissions*] **AND**
      radar image   = *snowy*
THEN DEDUCE
        intent   = *hostile.*

However P5 suggests that in some circumstances R2 may lead to precisely the wrong conclusion. The next logical query, therefore, is to determine the probability that such circumstances will occur. Once again probability propagation can be used to dedeuce

Prob(passive sensor 2 = *no emissions* | P6

              [passive sensor 1   = *strong emissions* ∨
               passive sensor 2   = *strong emissions*]
              radar image   = *snowy*
              general hostilities = *hostile*   )   = .03.

P6 indicates that if the preconditons of R2 are satisfied, then there is only a 3% change of being in the circumstance (identified in P5) where R2 ouputs a misleading result.

Continuing with this procedure a team designer can eventually construct a set of individual decision procedures, such as the set of decision rules shown in Figure 4.3.4.

IF radar image = *snow* AND
    [passive sensor 1 = *strong emissions*   OR passive sensor 2 = *strong emissions*]
THEN intent = *hostile*

IF passive sensor 1 = *no emissions* AND
    passive sensor 2 = *no emissions* AND
    radar track = *toward zz* AND
    [radar distance ≤ *500nm*   OR radar size > *1*]
THEN intent = *hostile*

IF passive sensor 1 = *no emissions* AND
    passive sensor 2 = *no emissions* AND
    [radar distance ≥ *500nm*   OR radar track = *away from zz*]
THEN intent = *possibly hostile*

IF radar image = *clear* AND
    NOT(passive sensor 1 = *no emissions*) AND
    NOT(passive sensor 2 = *no emissions*) AND
    radar distance ≥ *500nm*
THEN intent = *not hostile*

IF radar image = clear AND
    radar distance ≥ *500nm* AND
    [passive sensor 1 = *strong emmisions*   OR passive sensor 2 = *strong emmisions*]
THEN intent = *not hostile*

Figure 4.3.4 Compiled Decision Procedures Derived from Domain Model

<u>Distribute Decision Procedures</u> - To specify the team's decision procedures, the individual rules must be distributed among the team members. An initial distribution can be determined by inspecting the probability graph, the proposed decision rules and other information to determine a "natural" groupings of nodes. In the case of Figures 4.3.3, for instance, it makes sense (at least initially) to separate a radar and a passive sensors operator, since all the radar data is probably on a single display. If a two person team is assumed, the it may make sense to separate responsibility into a passive sensor operator, and a commander who also monitors the radar data This is depicted in Figure 4.3.5.

This depiction uses the Petri Net notation of Andreadakis and Levis (1987). In the case of Figure 4.3.5 there are two decision makers (DM1 and DM2). Each DM executes decision procedures that can be characterized as Sensor Assessment (SA), Information Fusion (IF), Command Interpertation (CI) or Response Selection (RS). DM1 is responsible for the passive sensors (SA-PS), while DM2 handles the radar system (SA-R). The bold lines indicate the active flow of tokens. The black rectangles correspond to active transitions, where the outputs do not equal the inputs. The darkened reactangles describe "pass through" transitions where the output equals the input. The other rectangles are inactive.

Once the architecture has been defined, the decision rules may need to be modified to accomodate the need to transmit information among the nodes. For instance many of the rules in Figure 4.3.4 could not be executed by either of the two agents in the architecture shown in Figure 4.3.5, since no one agent has direct access to both radar and passive sensor data. This implies that some of these rules must be further split apart, adding additional

variables must be introduced into the system. A modified set of rules is shown in Figure 4.3.6.
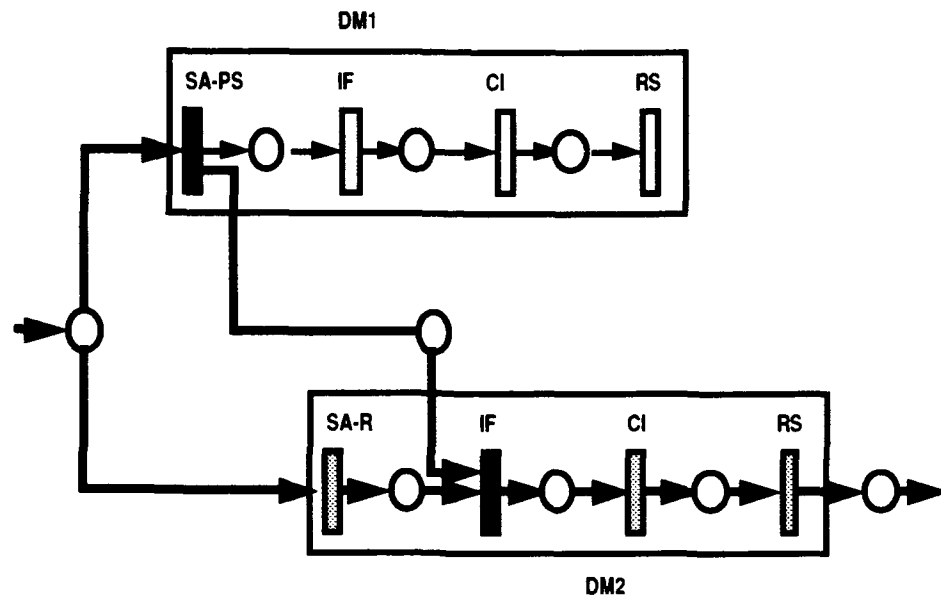

Figure 4.3.5 Proposed Distribution of Compiled Decision Procedures

<u>Evaluate Performance</u> - Together Figures 4.3.5 and 4.3.6 completely define a team's decision procedures. The structure of the team has been specified and specific decision procedures have been allocated to each team member. The question now arises as to whether or not this is a good team design. Below we describe how to evaluate a team design from several perspectives.

Deduce Accuracy of Decision Procedures - The product of this step is an *output accuracy table* such as Table 4.3.1. The columns are possible outputs of the team's decision procedures. The rows are the possible hypothesis states. The values along the rows indicate the conditional probability of each possible output of D given H. For instance, the upper left cell of Table 4.3.1 asserts that P("*hostile*"|intent=*hostile*) = .74. where "=*hostile*" means that the output of the team decision procedure is *hostile*. In general we use quotes here to indicate the output of a decision procedure (i.e.. "x" is the same as D|--x).

DM2: IF

IF radar image = *snow* AND
    passive sensors = *strong emissions*
THEN intent = *hostile*

IF passive sensors = *no emissions* AND
    radar track = *toward zz* AND
    [radar distance $\leq 500nm$ OR radar size >1]
THEN intent = *hostile*

IF passive sensors = *no emissions* AND
    [radar distance $\geq 500nm$ OR
       radar track = *away from zz*]
THEN intent = *possibly hostile*

DM1: SA-PS

IF passive sensor 1 = *strong emissions* OR
    passive sensor 2 = *strong emissions*
THEN passive sensors = *strong emissions*
ESLE
IF passive sensor 1 = *no emissions* AND
    passive sensor 2 = *no emissions*
THEN passive sensors = *no emissions*
ELSE passive sensors = *weak emissions*

IF radar image = *clear* AND
    passive sensors = weak emissions
    radar distance $\geq 500nm$
THEN intent = *not hostile*

IF radar image = clear AND
    radar distance $\geq 500nm$ AND
    passive sensors = *strong emmisions*
THEN intent = *not hostile*

Figure 4.3.6 Alternative Decision Procedures

These probabilities are assessed as follows. Let {hi} be a set of possible states, and {di} a set of possible outputs from a decision procedure. Then

$$P(dj|hi) = \Sigma_{ek} \; P(dj|e_k \& h_i)P(e_k|h_i) \qquad \text{eq. 1}$$
$$= \Sigma_{\{ek|D|--dj\}} \; P(e_k|h_i).$$

That is P(dj|hi) is equal to the sum of $P(e_k|h_i)$ for all evidential states e_k where $D|--d_j$. For small probability networks, these values can be calculated exactly using exact Bayesian update procedures. For larger models, forward simulation techniques can be used to approximate these values. The development of approapriate techniques is part of research task 4.5.

Elicit Utilities and Calculate Expected Utility - For each cell in the Output Accuracy Table a relative utility value can be elicited. This information can be used to calculate the expected utility of the decision procedures.

Evaluate Robustness - A team decision procedure is robust to the extent that it can be faithfully executed under adverse conditions. With regard to robustness execution. two issues are considered-- *workload* and *cognitive biases*. Task workload is measured in terms of the number and distribution of decision variables needed to execute the decision procedure. Although the general methodology does not require it. we are currently using the task workload measures defined in Boettcher and Levis (1983) and Andreadakis (1988) and used by Jin (1990). Here we are looking for a decision procedure where task workload is kept small.

TABLE 4.3.1 An Output Accuracy Table

OUTPUT OF DECISION PROCEDURES

	hostile	possibly hostile	not hostile	unknown	conflicting rules
hostile	.57	.21	.04	.11	.07
neutral	.02	.34	.41	.13	.10
friendly	.01	.07	.72	.07	.13

(INTENT labels the rows: hostile, neutral, friendly)

Cognitive biases refer to consistent deviation from normative judgments. It is well documented in the behavioral decision theory literature that people employ heuristic procedures to make judgments, and that these heuristic procedures consistently lead to less than optimal judgments. For instance, people often engage in a heuristic called anchoring and adjustment. They anchor on an initial value and adjust their assessment to account for additional information. Often people underestimate the value of the additional information and do not adjust enough. The tendency of people to consistently under adjust on the basis of new information is an example of a cognitive bias. A *vulnerable-to-bias* decision procedure is a decision procedure which provides the opportunity for biases to occur. We expect that vulnerable-to-bias decision procedures are more vulnerable to stress that other decision procedures.

Documentation

Lehner, P. "Towards a theory of team design." *Proceedings of the 1991 Symposium on Command and Control Research*, June 1991, 149-159.

Lehner, P. "Towards a prescriptive theory of team design." *Proceedings of the 1991 IEEE Conference on Systems, Man and Cybernetics*. September 1991, 2029-2034.

4.4 EXPERIMENTAL RESEARCH TO EVALUATE VULNERABLE-TO-BIAS DECISION PROCEDURES

Background. C2 teams are composed of a group of interacting decision makers working cooperatively to solve a common decision problem. Each team member has an area of expertise. Each team member is responsible for a distinct set of inference and decision functions for which each team members is well-trained. Under conditions of low stress, one would expect a well-trained team to reliably execute the procedures they have been taught and to perform well. An open question, however, is the extent to which training breaks down under conditions of high stress. Except for issues related to task workload, this issue has not been addressed.

The objective of this research task is to investigate the impact of cognitive biases on the performance of well trained teams under stress. Our research contrasts two perspectives.

Perspective 1 (P1) - Cognitive biases are largely a matter of preference. Although people tend to use heuristic rules that deviate from normative procedures, they can be taught to reliably use alternative rules, as long as the alternative rules do not exceed bounded rationality constraints.

Perspective 2 (P2) - Cognitive biases are largely a matter of capability. Even if trained, people do not reliably execute judgment and decision procedures that do not conform to cognitive biases.

For team decision making under stress, these two perspectives differ considerably with respect to their implication for designing teams. If P1 is correct, then the literature on human cognitive biases is simply irrelevant to the problem of designing teams. Properly trained and practiced teams will reliably execute correct decision procedures until workload or other bounded rationality constraints are exceeded. If P2 is the correct, then cognitive bias considerations should place severe constraints on the design of a team. Specifically, one should avoid specifying team architectures and decision procedures that are inconsistent with the heuristic decision making procedures that people naturally use. Otherwise, these teams will be vulnerable to cognitive biases, and the team's decision procedure will not be executed reliably under high stress conditions.

Several experiments investigating this issue will be performed.

Progress during this period. The first experiment was completed during this period. The analysis of the results will be completed during the next period.

Experiment 1. The first experiment investigated the impact of time stress on team decision making performance. Specifically, we were interested in the extent to which simple, but vulnerable-to-bias decision procedures would break down under time stress.

Method - The experiment was a modification of Jin's (1990). Two person teams worked together to defend a battle group from incoming aerial attacks. Each team member must assess the type of aircraft associated with each radar track. Figure 4.4.1 shows a screen display from DM1's (decision maker 1) perspective. All procedures and displays for DM2 are symmetric. For tracks in the DM1-only region (indicated on Figure 4.4.1), DM1 must click on the track, examine the information in the *Tactical Information Window*, select *Speed* and *Type* (below tactical display) and then select *Attack*. For tracks in the DM2-merge region, DM1 executes the same steps as in the DM1-only region, except that instead of selecting *Attack*, DM1 must send a message to DM2 indicating the selected type. For the DM1-merge region, DM1 executes the same steps as in the DM1-only region except that after *Type* is selected, any message that DM2 may have sent regarding the type of that track is immediately displayed in the *Message Window*. DM1 then has the option of revising the *Type* selection before selecting *Attack*.
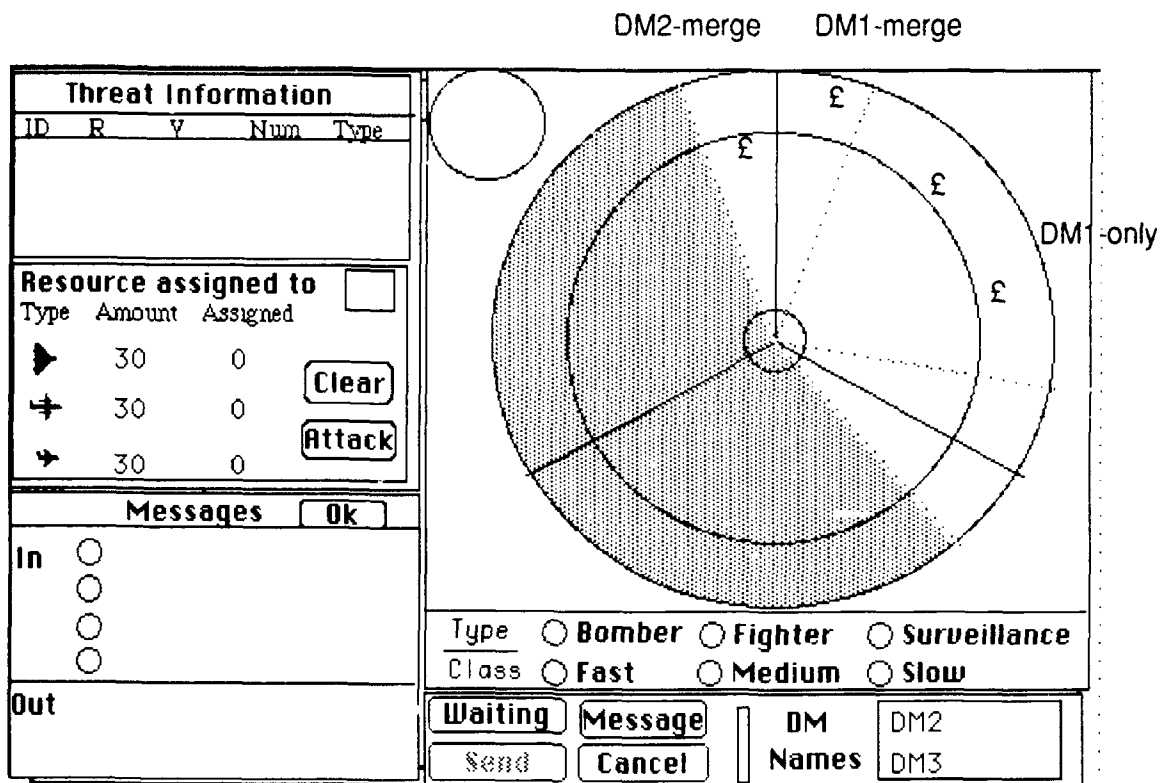
Figure 4.4.1   *Screen Display for Experiment 1.*

Teams are trained in the *Type* selection decision procedures shown in Figure 4.4.2. Note that some of the procedures in Figure 4.4.2b are potentially vulnerable to bias. Specifically, if DM1's initial judgment was F and DM2's "advice" is B (the FB condition), then DM1 should accept DM2's advice and change to B. This is also true for the FS condition. This procedure is potentially vulnerable to the following biases.

SITUATION ASSESSMENT RULES

Size

	<5	5	>5
<500	S	F	B
500-800	B	S	F
>800	B	S	F

(Speed on vertical axis)

a

INFORMATION FUSION RULES

DM2 Advice

	F	B	S
F	F	B	S
B	B	B	F
S	S	F	S

(DM1 Judgment on vertical axis)

b

Figure 4.4.2 Decision Procedures for Experiment 1

<u>Anchoring and Adjustment</u> - After anchoring on an initial judgment, people often under adjust their judgments in the light of new evidence. Consequently, one might expect DM1 to sometimes ignore the incoming advice.

<u>Confirmation Bias</u> - Once an initial judgment has been made, people often devalue or underweight the importance of new information that is contradictory to the initial judgment. Consequently, this bias might also lead DM1 to sometimes ignore the incoming advice.

<u>Attribution Error</u> - When faced with similar decision problems (and performance levels) people tend to attribute performance errors in other people the their poor decision making, but their own errors to the environment ("It couldn't be helped."). Consequently, one might anticipate that this bias would lead DM1 to undervalue DM2's performance and advice.

In addition, the BS and SB conditions are subject to the same problems as the FS and FB conditions, but here DM1 must select the hypothesis that neither team member mentioned, namely F. Although such procedures may be normative, they are counter to how people typically process advice and their own initial judgments.

Note that the vulnerable-to-bias procedures which were the subject of this study all relate to the interaction between team members. The research is focused on biases that are uniquely relevant to team decision making and coordination.

There were ten teams. Each team was trained for approximately 1.5 to 2 hours. After this each team prosecuted 120 trails. This took approximately 4 hours over a two day period. Each trial contained 8 incoming tracks. The speed of the incoming tracks on each trail was either 400 mph, 600 mph, 750 mph, 900 mph. In the 400 mph condition, a team had approximately 40 seconds to process the 6 tracks. For the 900 mph condition, a team had approximately 20 seconds to process all the tracks.

<u>Pilot Study</u> - Four teams were also run during a pilot study. This method was similar to the main experiment except that the lowest stress condition allowed subjects about 50 seconds to prosecute all the tracks. The results pilot subjects were similar to the main results with the exception of the lowest stress condition. This will be discussed below.

<u>Results of Experiment 1</u>. If P1 is correct, then performance under the BF,SF conditions should be equal to that of the FS,FB and BS,SB condition. If P2 is correct, then performance under the BF,SF condition should be higher than the other two conditions, particularly as time stress increases. Below we show a performance summary result from the ten teams from the first experiment. This gave us data for 20 subjects.

Table 4.4.1 shows the aggregate (total number correct/total number in cell) performance for each type of merged judgment at each stress level.

Table 4.4.1 Aggregate Proportion Correct for all Ten Teams

Speed	Judgments Agree (ff, bb,ss)	Disagree, but not vulnerable to bias (bf,sf)	Disagree and vulnerable to bias (fb, fs)	Disagree and vulnerable to bias (bs, sb)
400	.98	.92	.82	.80
600	.95	.89	.78	.85
750	.97	.93	.71	.79
900	.95	.75	.64	.63
Average	.96	.87	.74	.77

At all stress levels, performance for the vulnerable-to-bias procedures was lower than for procedures that were consisdered not vulnerble to bias. Interestingly, these results differ from those of the pilot study in that in the pilot study performance at the lowest stress level was between 95 and 100% for all the decision procedures.

TABLE 4.4.2
Number of Teams for which Performance was in
predicted direction, equal, or reverse direction.
(Includes one-tailed sign test results)

		Hypothesis	
Speed	**BF/SF > FB/FS**	**BF/SF > BS/SB**	
400	11,8,1 (p=.0032)	15,3,2 (p=.0012)	
600	10,6,4 (p=.0898)	10,5,5 (p=.1509)	
750	13,6,1 (p=.0009)	12,5,3 (p=.0176)	
900	8,9,3 (p=.1133)	11,7,2 (p=..0112)	
Overall	42,29,9 (p<.0001)	48,20,12 (p<.0001)	

Table 4.4.2 summarizes these results of a team by team analysis. Each cell indicates the number of teams for which the proportion of correct responses was in the expected direction, equal or reversed. For instance, for the cell in the first row and column, there were eleven, eight and one team(s) respectively for which the proportion correct for the BF/SF judgments were greater than, equal to, or less than proportion correct for the FB/FS judgments. In addition, the results are shown of a one-tailed sign test comparing the number of subjects for which the results were in the predicted versus the reverse direction. For instance, for a one tailed sign test, the probability of observing a ratio of 11 to 1 or greater is .0032. Overall, it is clear that BF/SF judgments are prosecuted more accurately than FB/FS and BS/SB judgments.

Discussion of Experiment 1. In general these results support the perspective that even if teams are well-trained, vulnerable-to-bias decision procedures are not robust to conditions of time stress. In this experiment, the decision procedures that subjects were taught were very simple. In the pilot study, under the lowest stress conditions, subjects executed these procedures almost without error. In the main experiment, subjects recei .ed slightly less training than in the pilot study and the lowest stress condition was more time stressed than in the pilot study. We were quite surprised to discover that these minor changes lead to substantial performance degradation for the vulneraᴸ‚e-to-bias procedures.

The principle implication of these results is that consideration of cognitive biases should be a major feature of the team design process. Even simple decision procedures are subject to these biases. Furthermore, it appears to be very difficult to train people out of these biases. One would expect that in low stress practice sessions, these biases would not manifest. However, as stress increases one would expect vulnerable-to-bias procedures to degrade quickly.

A report summarizing the results of the first experiment is being prepared. Completion is expected in May 1992.

Experiment 2 - The second experiment builds on the results of the first experiment. The second experiment will differ from the first experiment in several ways.

1. The decision procedures will be more complex than in the first experiment.

2. The focus will be on decision procedures related to team coordination. Specifically, we will examine the extent to which decision procedures related to shifting coordination strategies degrade under conditions of stress.

The second experiment will also be hosted on a new testbed. The testbed used for the first experiment was imported from MIT. It was specifically designed to support Jin's (1990) research. We found it very difficult to modify. We felt that in the long run, a great deal of time and effort could be saved by developing a new testbed that is designed to be easily modified. Work on this testbed began in July. Completion is expected in November.

In addition, a literature review of the cognitive biases literature was initiated to identify research results relating to training people out of cognitive biases. A report summarizing this review was prepared by Dr. Michael O'Conner.

Documentation

Lehner, P., Nallappa, B., O'Conner, M., Saks, S. and Mullin, T. "Cognitive Biases and Stress in Team Decision Making: Preliminary Report," *Proceedings of the 1991 BRG Symposium on Command and Control.*

O'Conner, M. "Cognitive Biases: A Perspective and Recommendations," GMU-C3I Center Technical Report (number pending), October 1991.

A technical report describing the first experiment is being prepared by Mr. Mir-Masood Seyed-Solorforough. Completion is expected in May 1992.

4.5   AUTOMATED TOOLS FOR SPECIFYING DECISION PROCEDURES.

Background As noted above, a domain model can be used to quantitatively evaluate a proposed set of team decision procedures. If necessary, this can be done manually using general purpose software for processing influence diagrams. However, this is an time consuming process, and there is no guarantee that near optimal decision procedures have been produced.

The objective of this task is to develop automated tools to *derive* team decision procedures from a domain model. The input to the tool will be a domain model and a proposed team architecture. The proposed architecture includes both the number of team members and the types of information flows among team members. The automated tools will tradeoff several factors in the specification of these procedures. These factors include expected performance, workload and cognitive biases.

Progress to date. We have acquired a software tool called IDEAL (INfluence Diagram Evaluation and AnaLysis) which provides the necessary functions for defining and exercising domain models. We have also identified specific procedures for evaluating a proposed set of decision procedures. Specifically, if we let $\{hi\}$ be a set of possible hypothesis states, and $\{di\}$ a set of possible outputs from a decision procedure, then the key problem is to derive the distribution $P(\{di\}|\{hi\})$. This is the distribution of outputs of the decision procedure given alternative hypothesis values. As it turns out

$$P(d_j|h_i) = \Sigma_{ek} \; P(d_j|e_k \& h_i)P(e_k|h_i)$$
$$= \Sigma_{\{ek|D|--dj\}} \; P(e_k|h_i). \qquad\qquad (Eq.\; 1)$$

That is $P(dj|hi)$ is equal to the sum of $P(e_k|h_i)$ for all evidential states e_k where Dl--d_j. For small probability networks, these values can be calculated exactly using exact Bayesian update procedures. However, for large and realistic problems $P(\{di\}|\{hi\})$ can be approximated using a forward simulation monte carlo procedure.

Using IDEAL, a simulation procedure was implemented by Mr. Seyed-Solorforough that approximates $P(\{di\}|\{hi\})$. This will be used for the second experiment.

Azar Sadigh has selected this area for her thesis research. She will implement a system that provides a set of tools to support network compilation. Completion of her thesis is expected in December 1992.

Documentation.

Lehner, P. and Sadigh, A. "Reasoning under uncertainty: Some monte carlo results." in *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference (1991)*, San Mateo: Morgan Kaufmann Publishers, 1991, 205-211.

4.6 QUANTITATIVE MODELS OF COMBINED USER/DECISION AID PERFORMANCE

Background. The literature on DSSs is replete with long lists of features of a "good" decision aid. Unfortunately, despite all this advice, there are very few models that proport to predict the effect that introducing a decision aid into an decision maker's setting will have on performance. This task will investigate the development of quantiative models of the impact of introducing a DSS into a team's decision process. This work is an extension of the result in Lehner, et. al. (1990) and the methodology described in research task 4.3 for deriving team decision procedures.

Progress to date. This work will be the Ph.D. Thesis of Mr. Thomas Lam under the supervision of Dr. Lehner. Mr. Lam is a program manager at DISA. He received a grant from DISA that will allow him to come to GMU full time in the Spring 1992 semester to pursue his disseration research. He will begin in February 1992. Completion of his thesis is expected in May 1992.

5.0 MEETINGS

In October, Dr. Lehner and Dr. Levis presented papers at the 1991 IEEE Conference on Systems, Man and Cybernetics at the University of Virginia.

6.0 RESEARCH PERSONNEL

6.1 Current Research Personnel

The following people are currently participating in this effort.

Prof Paul Lehner   GMU - Principal Investigator
Prof. Alexander H. Levis,   GMU

Mr. Mir-Masood Seyed-Solorforough	GMU - Graduate Research Assistant (Ph.D.)
Mr. Thomas Lam	GMU - Graduate Research Assistant (Ph.D.)
Mr. Syed Abbas K. Zaidi	GMU - Graduate Research Assistant (MS)
Mr. Zhuo Lu	GMU - Graduate Research Assistant (MS)
Ms. Azar Sadigh	GMU - Graduate Research Assistant (MS)
Mr. Steve Saks	DSC - Programmer

6.2 Previous Research Personnel

The following persons were previously supported by the research effort.

Mr. Bhashyam Nallappa	GMU - Graduate Research Assistant (M.S.)
Dr. Kent Hull	DSC
Dr. Martin Tolcott	DSC - Consultant
Dr. Theresa Mullin	DSC
Dr. Michael O'Conner	DSC - P.I. of subcontract
Mr. William Roman	DSC - Programmer
Dr. Michael Donnell	Consultant

6.3 Personnel Changes

As mentioned in previous progress reports, there has been a substantial turnover in personnel at DSC. Dr. Kent Hull was previously replaced by Dr. Michael O'Conner as the DSC project manager. Dr. O'Conner has now left DSC as well. Currently, there is no principal investigator at DSC. Dr. Tolcott was participating as a consultant to DSC. He is no longer working on this effort. Dr. Mullin was a principal contributor to the design of the first experiment, but was reassigned by DSC. She is no longer working on this effort. Mr. Roman was working as a programmer on the new experimental testbed. He has also left DSC during this period. When the current assignment is completed, the sub-contract to DSC will be reviewed in terms of the project needs.

7.0   DOCUMENTATION

7.1 Theses

1. Zaidi, S. A. K., "On the generation of Multilevel, Distributed Intelligence Systems using Petri Nets," MS Thesis, Report No.: GMU/C3I-113-TH. C3I Center, George Mason University, Fairfax, VA. November 1991.

2. MS Thesis by Z. Lu in preparation - due April 1992.

3. MS Thesis by A. Sadigh in preparation - due December 1992.

4. Ph.D. Thesis by T. Lam - due May 1993.

7.2 Technical Papers

1. Lehner, P., Nallappa, B., O'Conner, M., Saks. S. and Mullin. T. "Cognitive Biases and Stress in Team Decision Making: Preliminary Report." *Proceedings of the 1991 BRG Symposium on Command and Control.*

2. Lehner, P. "Towards a Theory of Team Design." *Proceedings of the 1991 Symposium on Command and Control Research.* June 1991, 149-159.

3. Lehner, P. "Towards a Prescriptive Theory of Team Design." *Proc. 1991 IEEE International Conference on Systems, Man and Cybernetics,* October 1991, 2029-2034.

4. Lehner, P. and Sadigh, A. "Reasoning under Uncertainty: Some Monte Carlo Results." in *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference (1991),* San Mateo: Morgan Kaufmann Publishers, 1991, 205-211.

5. Levis, A. H., "A Colored Petri Net Model of Intelligent Nodes" *Proc. 1991 IMACS Symposium on Modeling and Control of Technological Systems,* Lille, France, May 1991.

6. Levis, A. H., "A Colored Petri Net Model of Command and Control Nodes" to appear in *Command, Control, and Communications: Advanced Concepts and Paradigms,* Carl R. Jones, Ed., AIAA Press, Washington DC. Book to appear in late 1992.

7. Lu, Zhuo and A. H. Levis, "A Colored Petri Net Model of Tactical Decision Making." *Proc. 1991 Symposium on C2 Research,* National Defense University, Ft. McNair. Washington, DC, June 1991.

8. Lu, Zhuo and A. H. Levis, "A Colored Petri Net Model of Distributed Tactical Decision Making," to appear in *Proc. 1991 IEEE International Conference on Systems, Man, and Cybernetics,* October 1991.

9. O'Conner, M. "Cognitive Biases: A Perspective and Recommendations." GMU/C3I Center Technical Report (number pending). October 1991.

10. Lehner, P.E. "Empirical Evidence in Artificial Intelligence Research." GMU/C3I Center Technical Report (number pending). February 1992.

8. REFERENCES

Andreadakis, S.K. (1988) "Analysis and Synthesis of Decision Making Organizations" LIDS-TH-1740, Ph.D. Thesis, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Balbes, R. and Dwinger, P. (1974). *Distributive Lattices* University of Missouri Press, Columbia, Missouri.

Boetcher, K., and Levis, A.H.. (1982) "Modeling the Interacting Decisionmaker with Bounded Rationality," *IEEE Transactions on Systems, Man, and Cybernetics,* SMC-12(3), 334-344.

Demaël, J. J (1989). "On the Generation of Variable Structure Distributed Architectures." LIDS-TH-1869, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Genrich, H. and Lautenbach, K. (1981) "Systems Modeling with High-Level Petri Nets." *Theoretical Computer Science*, No.13, pp.109-136.

Jensen, K. (1987). "Colored Petri Nets" *in Advanced Course on Petri Nets 1986, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany.

Jin, V. Y. (1990) "Effect of Organizational Structure on Performance of Decision Making Teams," LIDS-TH-1976, Ph.D. Thesis, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Lehner, P.E., Mullin, T.M. and Cohen, M.S. "A Probability Analysis of the Usefulness of Decision Aids" in *Uncertainty in Artificial Intelligence: Volume 5*, Henrion, M., Schachter, R.,Kanal, L.N., and Lemmer, J. (eds.), Elsevier, North-Holland, 1990.

Levis, A. H. (to appear in 1992). "Modeling and Design of Distributed Intelligence Systems" in *Introduction to Autonomous and Intelligent Control*, P. J. Antsaklis and K. M. Passino, Eds., Kluwer Publishers, Boston, MA.

Mesarovic, M. D., Macko, D., and Takahara, Y. (1970). *Theory of Hierarchical, Multilevel Systems*. Academic Press, New York.

Monguillet, J.-M., and Levis A. H. (1988). "Modeling and Evaluation of Variable Structure Command and Control Organizations." *Proc.1988 Symposium on Command and Control Research.* Science Applications International Corporation, McLean, Virginia.

Remy, P. (1986). "On the Generation of Organizational Architectures Using Petri Nets." LIDS-TH-1630, Laboratory for Information Decision Systems, MIT, Cambridge, MA.

Remy, P. and Levis, A. H. (1988). "On the Generation of Organizational Architectures Using Petri Nets." in *Advances in Petri Nets 1988, Lecture Notes in Computer Science*, G.Rozenberg Ed. Springer-Verlag, Berlin, Germany.